

Where do goals come from? A Generic Approach to Autonomous Goal-System Development

Matthias Rolf and Minoru Asada

Abstract—Goals express agents’ intentions and allow them to organize their behavior based on low-dimensional abstractions of high-dimensional world states. How can agents develop such goals autonomously? This paper proposes a detailed conceptual and computational account to this longstanding problem. We argue to consider goals as high-level abstractions of lower-level intention mechanisms such as rewards and values, and point out that goals need to be considered alongside with a detection of the own actions’ effects. We propose *Latent Goal Analysis* as a computational learning formulation thereof, and show constructively that any reward or value function can be explained by goals and such self-detection as latent mechanisms. We first show that learned goals provide a highly effective dimensionality reduction in a practical reinforcement learning problem. Then, we investigate a developmental scenario in which entirely task-unspecific rewards induced by visual saliency lead to self and goal representations that constitute goal-directed reaching.

Index Terms—Latent Goal Analysis, Goal Systems Development, Self Detection, Recommender Systems, Dimension Reduction, Goal Babbling, Saliency

I. MOTIVATION

Goals are abstractions of high-dimensional world states that express intelligent agents’ intentions underlying their actions. Goals are considered to organize the behavior of both humans and robots. For instance in robot *planning* [1] as well as motor *control* [2], [3] goals describe the desired outcome of future actions in terms of what aspect or variable in the world is relevant and what its desired value is. Also in robot *learning* goals have been proven useful as a scaffolding mechanism to perform efficient exploration also in high dimensional sensorimotor spaces [4], [5]. Yet, in all of these scenarios the goals are carefully handcrafted: the variable to be controlled, as well as a mechanism to select a particular goal of the agent at any time need to be specified by the designer. Several formulations of motor learning can automatically choose internal goals purely for the sake of training a skill (e.g. [6], [5]), but they do neither explain how to choose goals depending on external stimuli, nor can they determine the variable that has to be controlled.

Goals are a fundamental concept also in neuroscience and psychology. The entire formulation of the cerebellum providing *internal* forward and inverse *models* [7] only makes sense if goals are already given as input for the inverse models. From a conservative standpoint such models concern motor control

in the first place. Recent theories, however, go much further and suppose that they also contribute to cortex-wide higher-level cognitive processes [8] and are engaged in social behavior [9]. Goals are also seen as a major factor in motivation psychology [10], where “goal-setting” is considered essential for long term behavior organization. Goals are considered a likewise structuring element in our cognition and perception of other agents’ behavior such as in imitation learning [11] or teleological action understanding [12], [13].

Goals are useful abstractions. But where do they come from? Neither robotics and machine learning, nor neuroscience, nor psychology provide conclusive or even general hints how a biological or artificial agent that starts with no goals can acquire them. Already Weng *et al.* [14] in 2001 pointed out that developmental robots should learn without a particular task (i.e. explicit goals) given to them at first. But then, how could they ever develop such abstractions? The consequent need to consider the learning *of* goals was explicitly pointed out firstly by Prince *et al.* [15] in 2005, but is essentially unsolved ever since. It may not be difficult to think of heuristics for an agent to acquire goals within isolated special scenarios, but what could be general mechanisms for a development of goal systems? This article seeks for answers to this longstanding question. We thereby focus on (i) the learning of an agent’s *own* goals, in contrast to observational learning about others’ goals such as in imitation learning [16], [17] or values such as in inverse reinforcement learning [18], and (ii) a fully *autonomous* learning without external supervision such as an agent being told what to do.

A. Overview

Our paper makes four contributions along with the next four sections. Firstly, we develop a general *conceptualization* of goals in Sec. II. We introduce several novel arguments in order to define goals as precisely as possible, and argue to consider goal systems as abstractions of reward systems. These arguments are meant to stimulate a wider discussion and provide the basis for our second contribution: In Sec. III we introduce a computational learning formulation *Latent Goal Analysis* (LGA). Based on given rewards we show how “latent” goals can be extracted from the sensory and action information and constructively prove the universal existence of this transformation. Thirdly, we show an *application* oriented experiment in Sec. IV. We use LGA as a dimension reduction technique for reinforcement learning. In a “recommender” scenario online news articles have to be recommended to readers. We use goals learned by LGA as compact representations of high-dimensional data that outperform standard methods

Matthias Rolf and Minoru Asada are with the Dep. Adaptive Machine Systems at Osaka University, Japan. Mail: {matthias, asada@ams.eng.osaka-u.ac.jp}.

This study has been supported by the JSPS Grant-in-Aid for Specially promoted Research (No. 24000012). This research is related to the European Project CODEFROR (PIRSES-2013-612555). We also would like to thank the Yahoo! Webscope program for providing the R6B - Yahoo! Front Page Today Module User Click Log Dataset, version 2.0 for our experiments.

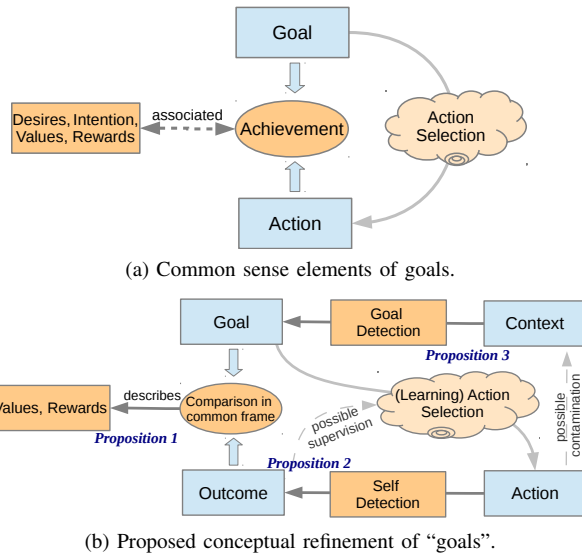


Fig. 1: We argue to consider goals as abstractions of rewards and values.

applied in that field. Lastly, we show an experimental setup to investigate the human development of goals for the case of reaching in Sec. V. We show that this task-unspecific information seeking reward based on visual saliency leads to the representation of to-be-reached objects as goals and a self-detection of the own hand. We thereby not only learn those abstractions, but already utilize them by applying goal babbling [19] to generate actions in a fully bootstrapping and closed action-perception-learning loop.

II. WHAT IS A GOAL?

How can we operationalize the acquisition of goals? In order to achieve a general conceptualization we start from common sense as in dictionary definitions and relate that meaning to usages in various scientific fields. Starting from this general definition we distinguish goals from several related terms such as optimization or affordances. Most importantly, we point out several vague or entirely unspecified aspects in the common sense usage of “goals” and make propositions how to substantiate the terminology. We thereby aim at a conceptualization that is both as general as possible and concrete enough for a mathematical operationalization.

A. General Definition and Related Terms

Dictionaries refer to the term “goal” as “an aim or desired result” of someone’s “ambition or effort”¹. Goals are most precisely defined in computational domains that use them. In motor coordination and control [2], [3] goals are typically low-dimensional abstractions of the task such as a desired angular velocity of an electric motor or a desired position of a robot’s end effector. Thereby goals are formulated as *values* in some *low-dimensional space* (e.g. 1d velocities, 3d positions) in which they abstract from many task-irrelevant variables (such

as room temperature) of the much higher-dimensional physical processes. Also goals in planning [1] describe world variables that should have some desired value, while other variables are irrelevant. In both control and planning the goal has to be *achieved* by means of *action*, i.e. the the agent has to find actions that result in the observation of the desired variable values. Similar aspects can also be found in goal-setting psychology [10]: For instance in management psychology it has been proposed that goals should be specific (have a particular value), measurable, and realistic (i.e. achievable by means of own action) [20]. The above points clearly distinguish goals from two other kinds of desires or intentions: (i) *Optimization* or general improvement (such as increasing reward) are not goals in a narrow sense. “Improvement” for itself is not specific in the sense that a particular to be achieved value is specified. Hence, there is no definite achievement possible or an end defined. (ii) *Wishes* of desired world states (e.g. having a sunny say) are no goals because they are not achievable by means of own action in the first place.

With these aspects we can attempt a first definition that we will refine in the remainder of this section:

Definition: A *goal* is an equivalence set of world states that, in a certain situation, an agent desires to achieve as a result of its own action.

They refer to an equivalence set of states in the sense that there can be irrelevant variables that do not matter for the goal. Hence, any of their values are equivalently acceptable. The main point is that goals reflect a particular desire. Their achievement has some value to the agent. This stands in contrast e.g. to *affordances* [21]. Sahin *et al.* formulated affordances as the relation between the action of an agent, an object under manipulation, and the effect on that object [22]. Objects with similar action-effect relations can then be summarized in equivalence classes such as “standonable”. Related to this formulation, *contingencies* [23], [24] and action-effect bindings [25], [26] describe general relations between actions and their specific effects. Goals and affordances are both interactivist concepts that can not be defined by only the agent or only the environment, but only via their interplay. The crucial difference between them is that affordances describe any *possible* thing that could be done. Affordances are not associated to any value or *desire*, while this desire to do something is the constituting aspect of goals in our view.

B. A Refined Concept of Goals

Intelligent organisms can not arbitrarily regard things as goals while others are not regarded as goals. The biggest open question about goals in our view is *what kind of information could lead to this differentiation between goals and non-goals*. This source of information, or learning signal, is consequently the main point of our overall argumentation. In terms of machine learning we know three basic kinds of learning signals: *supervised* input of ground-truth values, *unsupervised* learning of input statistics, and *reward* or cost signals in reinforcement learning or optimization. Supervised learning as source of information seems entirely unsuited for autonomous development of goals, since a teacher for such information

¹oxforddictionaries.com “goal”; corresponding definitions in other languages: duden.de (German) “Ziel”; nlpwww.nict.go.jp/wn-ja/ (Japanese) Synset 05980875-n; queried 2014/01/15

would have to be external. While social learning of goals in such terms certainly exists, it does not provide answers for an ontogenetic core mechanism of goal systems development. Unsupervised learning seems likewise unsuited since signal statistics can not tell about a desire or value. Rather, reward signals seem to be the suitable learning signal, as they express the most primitive form of a value. Considering goals as high-level abstractions of intention therefore suggests to consider them *abstractions of world states* that are associated to reward:

Proposition 1: Goals are *abstractions* that do not themselves *determine* a desire, but rather *describe* it based on lower-level systems of desire, such as reward or value systems.

Corresponding rewards might reflect some task very directly when e.g. determined in a social context, or directly as food. However, they might also be purely *internal* or *intrinsic* as it is often considered in the contexts of intrinsic *motivation* [27], [28], [29], [30] or information seeking [31]. Our exemplary experiment in Sec. V will take the latter perspective. The abstraction process thereby could not only concern immediate rewards, but also expected, future rewards that are expressed in *value systems* (e.g. supposed to exist in the midbrain [32]). In Fig. 1b we illustrate this proposition by saying the achievement of a goal (the abstraction) “describes” an (expected) reward.

The second question we must clarify to conceptualize goals is *what “achievement” means*. In natural language it is often expressed that an “action achieves a goal”. However, this skips an important aspect that must be considered for a profound conceptualization: Goals do not come alone, but always paired with an *evaluation* of the own action’s effect. It is the *effect* of the action, not the action itself to which the goal is compared. In robot reaching this evaluation, or rather its learning, is often referred to as *self-detection* [33], [34]: the robot’s hand needs to be detected for instance in a camera image. Just in the case of reaching, this notion is related from “body schema” [35] that describe a localization of the body in general. Here we use the term self-detection also in a non-reaching and non-spatial sense that describes any effect related to a goal. For instance the goal of having a sandwich to eat (e.g. followed by the verbal action “make me a sandwich”) would be accompanied by the detection of actually having a sandwich or not. The evaluation of the action equally exists in planning (e.g. navigation) domains, where the effect of past action is compared to the desired outcome, and goal-setting psychology, where a major emphasis is made on measurability of goals. In all of these domains the effect of the own action (self-detection) is then compared to the goal within a common reference frame in order to assess the achievement. The need for this comparison forbids considering the development of goals and self-detection separately from each other:

Proposition 2: Goals cannot be learned or considered independently from *self-detection*, but both have to describe a consistent *reference frame* in which the goal can be compared to the outcome of an action.

This aspect will largely guide our computational formalization. Our experiments will also illustrate that this aspect poses an important developmental hallmark: the entrance of *self-*

supervised action and motor learning. Once self-detection and goals are available, a supervised learning signal becomes available to other learning processes. When self-detection (e.g. a robot’s forward function) and goals were already available they have been used in numerous approaches for motor learning already [3]. With respect to the autonomous development of goals already Prince [15] noted that goals are related to self-supervision, but missed the point that it is not the goals themselves, but rather the self-detection (in relation to the goals) that enables the supervision.

Finally, we need to consider how goals become “active”, i.e. how an agent determines which goal to follow at a present moment. It is often considered that agents can have multiple goals, e.g. on different timescales or also parallel or secondary goals in the long run. This leaves a lot of play for an operationalization, which we propose to organize with the following restriction using the notion of cognitive or processing “(sub-)systems” internal to an agent:

Proposition 3: One system can have only *one* (active) goal *at a time*, which expresses the *present* desire based on an internal or sensory *context*.

Hence, different systems of cognitive processing (e.g. organizing different timescales of behavior) or motor planning and control can have one present goal each. Further we note that a goal gets triggered by a *context*, such as sensory information, an internal state or information from other processing systems. This seems trivial but makes an important point: there needs to be a *goal-detection* to determine the now to be followed goal from the context, and that parallels the self-detection. In the reaching example this might be a mechanism to determine the position of the relevant object from camera images, or also hard-wired position selectors in many robot setups. In terms of motivational psychology, for instance a certain context of a conversation might trigger a subsequent communicational goal such as to convey an information (“by the way...”).

In summary, we refer to a *goal system* as the joint apparatus of goal-detection from a context and self-detection that are compared within a common reference frame, such that the achievement of the goal by means of own action is reflected by a reward or value (see Fig. 1b).

III. LATENT GOAL ANALYSIS

In the last section we have established a conceptual framework of goals and argued to learn them together with self-detection as abstractions of reward or value signals. In order to develop a mathematical learning framework, we now have to set those terms in a *formal* relation. The best way to do so is to consider a domain which *already* has a formal relation of at least most of the conceptual elements, such as *control* or *coordination* problems [2], [3], [4]. In this section we develop our computational learning framework for goals by introducing this set of formal relations, adding terms from the concept that are still missing (Sec. III-A), and then establishing a learning rule from rewards to mathematical terms corresponding to each of the conceptual terms (Sec. III-B). In order to establish a generic framework we will not make any assumptions about the source of rewards, but will show examples for extrinsic (Sec. IV) and intrinsic (Sec. V) rewards in the experiments.

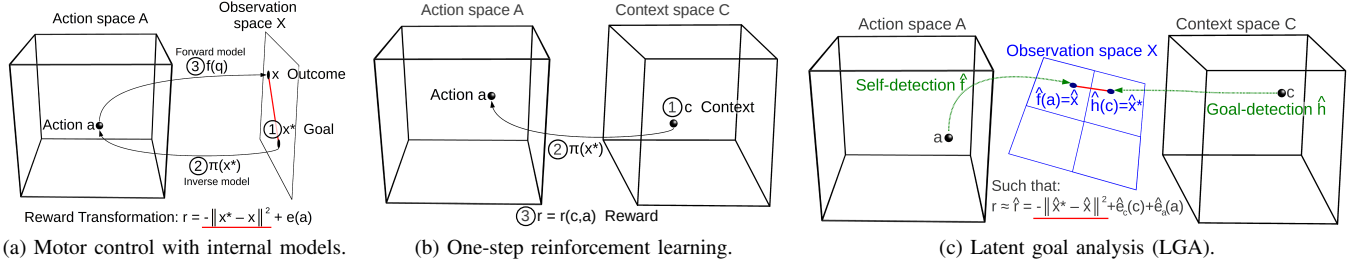


Fig. 2: Latent goal analysis (LGA) identifies how to project actions and contexts into a common observation space. The observed rewards r are thereby explained by distance between action-outcome \mathbf{x} (self-detection) and goal \mathbf{x}^* (goal-detection), such that a reward problem is turned into a control problem.

It can be argued that starting from such a particular domain as motor control considerably narrows the scope of the overall concept. However, we will show in this section the *universality* of this mathematical approach, i.e. that it can learn goals from *any reward or value function*. Specifically, we show an equivalence relation between motor control and reinforcement learning, such that any reinforcement learning problem can be transformed into a motor control problem with its abstractions for goals and self-detection. Hence, our learning framework is as general as reinforcement learning, which is often considered the most general learning formulation at all.

A. Reward Transformation

We start by establishing a formal relation of the conceptual terms based on motor control or coordination problems used in contexts of robot control [2] and learning based on internal models [7], [3], [19], [5]. Motor control problems as shown in Fig. 2(a) follow a simple protocol: (1): The world provides a goal \mathbf{x}^* to the agent that is situated in some observation space $\mathbb{X} \subseteq \mathbb{R}^n$. (2): The agent chooses an action \mathbf{a} from some action space $\mathbb{A} \subseteq \mathbb{R}^m$. (3): The world provides a causal outcome $f(\mathbf{a}) = \mathbf{x}$ of the agent's action, again situated in \mathbb{X} . The agent's task is to choose an action such that the outcome \mathbf{x} matches the goal \mathbf{x}^* : $\mathbf{x} = f(\mathbf{a}) = \mathbf{x}^*$. Many coordination problems provide *redundancy*: the action space is substantially higher dimensional than the observation space ($n \ll m$), such that multiple actions $\mathbf{a}_i \neq \mathbf{a}_j$ map to same outcome $f(\mathbf{a}_i) = f(\mathbf{a}_j)$. In such scenarios additional cost functions $-e_a(\mathbf{a})$ are often considered [2] to select an optimal action among all those that fulfill $f(\mathbf{a}) = \mathbf{x}^*$. In the case of hand-eye coordination \mathbf{a} could correspond to joint angles or torques. $\mathbf{x} = f(\mathbf{a})$ is the position of the end-effector that results from applying \mathbf{a} . The ground truth function $f: \mathbb{A} \rightarrow \mathbb{X}$ is called *forward function*. When this function is learned from supervised examples it is called *forward model*. The task to identify it (without supervision) is called *self-detection* [34]. The agent's selection of an action is addressed with an *inverse model* (see Fig. 2(a)) that is an inverse function of the forward model [7], [36]. This problem is often considered by means of an overall cost-function of the distance of goal and outcome, and $e_a(\mathbf{a})$, which is easily transformed into reward semantics by inverting the sign:

$$r(\mathbf{x}^*, \mathbf{a}) = -\|\mathbf{x}^* - f(\mathbf{a})\|^2 + e_a(\mathbf{a}) \quad (1)$$

What is still missing in this formulation is the goal-detection. This mechanism is not usually part of academic papers about motor control, yet always present: The goals of control processes are chosen by vision processes identifying relevant objects to be manipulated, planning- or other processes that are usually hard-wired. Hence, they are in some way determined by a larger internal or external context. We can denote this selection on an abstract level with a function $h(c)$, which we refer to as *goal-detection*. Further, we introduce for the sake of symmetry a virtual cost term $e_c(c)$ that only depends on the context, and which we will later need for our theoretical considerations. This term does not influence the optimal action selection for a given context, but introduces the aspect that the optimally possible reward depends on the context. Altogether this gives the reward transformation

$$r(\mathbf{c}, \mathbf{a}) = -\|h(\mathbf{c}) - f(\mathbf{a})\|^2 + e_c(\mathbf{c}) + e_a(\mathbf{a}) . \quad (2)$$

The overall protocol now corresponds to a one-step *reinforcement problem* [37], [38] as shown in Fig. 2(b): (1): The world provides a context \mathbf{c} in some context space $\mathbb{C} \subseteq \mathbb{R}^p$. (2): The agent chooses an action \mathbf{a} from the action space $\mathbb{A} \subseteq \mathbb{R}^m$. (3): The world provides a reward $r \in \mathbb{R}$ based on latent goals and action outcomes as in equation 2.

B. Latent Goal Transformation

We have now established a formal relation between all terms in our concept. Yet, so far this formulation only allows to transform existing goals into rewards. The central idea for computational learning of goals is now to *invert* this process. Given *any possible* reward or value function, is it possible to transform in *back* into a motor control problem?

In fact, this should not be possible according to common beliefs about reinforcement learning: Motor control comes with highly informative and usually low-dimensional abstractions that can be used for supervised learning [7], [3]. Goal and actual values in motor control define a relation similar (see [36]) to actual and target outputs in classical supervised learning setups by providing “directional information” in contrast to a mere “magnitude of an error” in reinforcement learning [39]. Given this rich structure in motor control, reinforcement learning seems the by far more general setup [40], [41].

Yet, we now show that in fact any reinforcement problem can be turned into a motor control problem, and thereby goals and self-detection can be retrieved as abstractions from a

reward signal. Therefore we need to find functions \hat{f} , \hat{h} , \hat{e}_c and \hat{e}_a to resemble any possible reward function $r(\mathbf{c}, \mathbf{a})$:

$$r(\mathbf{c}, \mathbf{a}) = \hat{r}(\mathbf{c}, \mathbf{a}) = -\|\hat{h}(\mathbf{c}) - \hat{f}(\mathbf{a})\|^2 + \hat{e}_c(\mathbf{c}) + \hat{e}_a(\mathbf{a}), \quad (3)$$

or value function $Q(\mathbf{c}, \mathbf{a})$ expressing expected future rewards:

$$Q(\mathbf{c}, \mathbf{a}) = \hat{Q}(\mathbf{c}, \mathbf{a}) = -\|\hat{h}(\mathbf{c}) - \hat{f}(\mathbf{a})\|^2 + \hat{e}_c(\mathbf{c}) + \hat{e}_a(\mathbf{a})$$

This work does *not* tackle the temporal credit assignment problem to estimate Q itself. However, if a value system [32], [42] to estimate future rewards Q is already available, decomposing either $r(\mathbf{c}, \mathbf{a})$ or $Q(\mathbf{c}, \mathbf{a})$ is computationally equivalent since both are scalar functions of \mathbf{c} and \mathbf{a} .

The major challenge is to identify the forward model by self-detection $\hat{f}(\mathbf{a}) = \hat{\mathbf{x}}$, and the goal-detection $\hat{h}(\mathbf{c}) = \hat{\mathbf{x}}^*$. Thereby goals and outcomes are considered *latent variables* of the reward function. These abstractions constitute the control problem by describing the interaction of goals and outcomes in a low-dimensional observation space (see Fig. 2(c)). Cost terms depending on context *or* action only are considered as remainders, and in fact are easy to find given \hat{f} and \hat{h} .

a) *Ansatz*: Finding such functions can be formulated as finding appropriate coefficients of parametrized functions. First, we consider features $\psi_c(\mathbf{c}): \mathbb{C} \rightarrow \mathbb{R}^{p'}$ and $\psi_a(\mathbf{a}): \mathbb{A} \rightarrow \mathbb{R}^{m'}$ to describe the contexts and actions. Assuming an n -dimensional observation space \mathbb{X} we can denote our function candidates with coefficients \mathbf{M} , \mathbf{H} , \mathbf{R}_a , and \mathbf{R}_c as:

$$\begin{aligned} \hat{\mathbf{x}}^* = \hat{h}(\mathbf{c}) &= \mathbf{H} \cdot \Psi_c(\mathbf{c}), \quad \mathbf{H} \in \mathbb{R}^{n \times p'} \\ \hat{\mathbf{x}} = \hat{f}(\mathbf{a}) &= \mathbf{M} \cdot \Psi_a(\mathbf{a}), \quad \mathbf{M} \in \mathbb{R}^{n \times m'} \\ \hat{e}_c(\mathbf{c}) &= \Psi_c(\mathbf{c})^T \cdot \mathbf{R}_c \cdot \Psi_c(\mathbf{c}), \quad \mathbf{R}_c \in \mathbb{R}^{p' \times p'} \\ \hat{e}_a(\mathbf{a}) &= \Psi_a(\mathbf{a})^T \cdot \mathbf{R}_a \cdot \Psi_a(\mathbf{a}), \quad \mathbf{R}_a \in \mathbb{R}^{m' \times m'}. \end{aligned}$$

When we insert these definitions into Eqn. 3 we can write the reward transformation of a control problem in matrix notation

$$\hat{r}(\mathbf{c}, \mathbf{a}) = \begin{pmatrix} \psi_c(\mathbf{c}) \\ \psi_a(\mathbf{a}) \end{pmatrix}^T \begin{pmatrix} \mathbf{R}_c - \mathbf{H}^T \mathbf{H} & \mathbf{H}^T \mathbf{M} \\ \mathbf{M}^T \mathbf{H} & \mathbf{R}_a - \mathbf{M}^T \mathbf{M} \end{pmatrix} \begin{pmatrix} \psi_c(\mathbf{c}) \\ \psi_a(\mathbf{a}) \end{pmatrix} \quad (4)$$

as a quadratic form of context- and action-features.

b) *Observation Space Reconstruction*: We can now write the *actual* reward function $r(\mathbf{c}, \mathbf{a})$ in a similar form by

$$\begin{aligned} r(\mathbf{c}, \mathbf{a}) &= \begin{pmatrix} \psi_c(\mathbf{c}) \\ \psi_a(\mathbf{a}) \end{pmatrix}^T \cdot \mathbf{K} \cdot \begin{pmatrix} \psi_c(\mathbf{c}) \\ \psi_a(\mathbf{a}) \end{pmatrix} \\ &= \begin{pmatrix} \psi_c(\mathbf{c}) \\ \psi_a(\mathbf{a}) \end{pmatrix}^T \begin{pmatrix} \mathbf{K}_{c,c} & \mathbf{K}_{c,a} \\ \mathbf{K}_{a,c}^T & \mathbf{K}_{a,a} \end{pmatrix} \begin{pmatrix} \psi_c(\mathbf{c}) \\ \psi_a(\mathbf{a}) \end{pmatrix}. \end{aligned} \quad (5)$$

This form with a symmetric coefficient matrix \mathbf{K} is a *universal approximator*: it can arbitrarily well approximate at least all continuous functions if appropriate features ψ are chosen. For instance, if the features ψ_a and ψ_c are separate polynomial features of \mathbf{a} and \mathbf{c} up to polynomial degree d , then just the subterm $\psi_c(\mathbf{c})^T \cdot \mathbf{K}_{c,a} \cdot \psi_a(\mathbf{a})$ will contain all joint polynomial terms of \mathbf{a} and \mathbf{c} up to degree d . Hence, equation 5 can at least describe all functions that can be described by polynomials, which are at least all continuous functions.

We can now find coefficients \mathbf{M} , \mathbf{H} , \mathbf{R}_a , and \mathbf{R}_c by matching equations 4 and 5. Starting from the need to match $\mathbf{K}_{c,a} = \mathbf{H}^T \mathbf{M}$, we can see that it is not only *always possible*

to transform rewards into goals and outcomes, but it is even under-determined. There are infinitely many decompositions $\mathbf{H}^T \mathbf{M}$ for any matrix $\mathbf{K}_{c,a}$. For any choice of \mathbf{H} and \mathbf{M} , a perfect match $r = \hat{r}$ can be generated by the residual terms

$$\mathbf{R}_c = \mathbf{K}_{c,c} + \mathbf{H}^T \mathbf{H} \quad \text{and} \quad \mathbf{R}_a = \mathbf{K}_{a,a} + \mathbf{M}^T \mathbf{M}.$$

For a concrete decomposition of $\mathbf{K}_{c,a}$ we can consider its singular value decomposition \mathbf{USV}^T with orthonormal matrices \mathbf{U}, \mathbf{V} and a positive diagonal matrix \mathbf{S} . An exemplary decomposition could be to set $\mathbf{H} = \mathbf{S}^{\frac{1}{2}} \mathbf{U}^T$ and $\mathbf{M} = \mathbf{S}^{\frac{1}{2}} \mathbf{V}^T$. Still, the resulting observation space \mathbb{X} , in which $\hat{f}: \mathbb{A} \rightarrow \mathbb{X}$ and $\hat{h}: \mathbb{C} \rightarrow \mathbb{X}$ map actions and contexts, is very high-dimensional with $\min(p', m')$ dimensions, since $\mathbf{K}_{c,a} \in \mathbb{R}^{p' \times m'}$. However, a dimension reduction is now straightforward based on the SVD: we can select the diagonal matrix $\mathbf{S}' \in \mathbb{R}^{n \times n}$ with the n largest singular values of $\mathbf{K}_{c,a}$ and their respective singular vectors in $\mathbf{U}' \in \mathbb{R}^{p' \times n}$ and $\mathbf{V}' \in \mathbb{R}^{m' \times n}$ to approximate $\mathbf{K}_{c,a} \approx \mathbf{U}' \mathbf{S}' \mathbf{V}'^T = \mathbf{H}^T \mathbf{M}$. $\mathbf{H} \in \mathbb{R}^{n \times p'}$ and $\mathbf{M} \in \mathbb{R}^{n \times m'}$ can be chosen within the column space of \mathbf{U}' and \mathbf{V}' in order to project into the n -dimensional observation space. Hence, the latent observation space can be uniquely determined for any number of dimensions n , except for multiple identical singular values. For sufficiently large n , LGA then approximates the reward function arbitrarily well.

c) *Optimal Self- and Goal-Detection*: While the selection of the observation space is uniquely determined by the column space of \mathbf{U}' and \mathbf{V}' , the positioning of goals and outcomes *inside* that space, i.e. the precise choice of \mathbf{H} and \mathbf{M} , is not unique. We can choose both matrices by means of any transformation matrices $\mathbf{P}_c, \mathbf{P}_a \in \mathbb{R}^{n \times n}$

$$\mathbf{H} = \mathbf{P}_c \cdot \mathbf{U}'^T, \quad \mathbf{M} = \mathbf{P}_a \cdot \mathbf{V}'^T, \quad \text{that satisfy}$$

$$\mathbf{P}_c^T \mathbf{P}_a = \mathbf{S}' \Rightarrow \mathbf{H}^T \mathbf{M} = \mathbf{U}' \mathbf{P}_c^T \mathbf{P}_a \mathbf{V}'^T = \mathbf{U}' \mathbf{S}' \mathbf{V}'^T.$$

We can see our options by representing \mathbf{P}_a by its singular value decomposition with orthonormal matrices $\mathbf{u}, \mathbf{v} \in \mathbb{R}^{n \times n}$ and positive diagonal matrix $\mathbf{s} \in \mathbb{R}^{n \times n}$:

$$\mathbf{P}_a = \mathbf{u} \mathbf{s} \mathbf{v}^T \Rightarrow \mathbf{P}_c = \mathbf{u} \mathbf{s}^{-1} \mathbf{v}^T \mathbf{S}'.$$

Using this notation it is easy to show that \mathbf{u} is irrelevant. If we insert the above definitions into the reward equation 4, we can see that \mathbf{u} only appears as $\mathbf{u}^T \mathbf{u}$. Since \mathbf{u} is orthonormal we get $\mathbf{u}^T \mathbf{u} = \mathbf{1}_n$ in any case and can set $\mathbf{u} = \mathbf{1}_n$ right away. This reflects that a rotation of the entire observation space does not change distances in that space, and therefore does not matter for LGA. It remains to choose an orthonormal matrix $\mathbf{v} \in \mathbb{R}^{n \times n}$ and a positive diagonal matrix $\mathbf{s} \in \mathbb{R}^{n \times n}$. \mathbf{v} and \mathbf{s} together determine how actions and contexts are precisely projected into the observation space as goals $\hat{h}(\mathbf{c}) = \hat{\mathbf{x}}^*$ and action-outcomes $\hat{f}(\mathbf{a}) = \hat{\mathbf{x}}$. Interestingly, $\hat{\mathbf{x}}^*$ and $\hat{\mathbf{x}}$ are thereby scaled “against” each other: Increasing \mathbf{s} results in scaling *up* $\hat{\mathbf{x}}$, but scaling *down* $\hat{\mathbf{x}}^*$ because of the appearance of \mathbf{s}^{-1} in \mathbf{H} . This operation largely modifies the distance between $\hat{\mathbf{x}}^*$ and $\hat{\mathbf{x}}$, and therefore the value of $-\|\hat{\mathbf{x}}^* - \hat{\mathbf{x}}\|^2$. Nevertheless this does *not* change the overall reward function \hat{r} , but shifts parameter “mass” between its different terms $-\|\hat{h}(\mathbf{c}) - \hat{f}(\mathbf{a})\|^2$, $\hat{e}_c(\mathbf{c})$, and $\hat{e}_a(\mathbf{a})$. Therefore also the contribution of rewards to \hat{r} by these different terms is changed by the choice of \mathbf{v} and \mathbf{s} . Of

course, $-||\hat{\mathbf{x}}^* - \hat{\mathbf{x}}||^2$ is the most relevant term in LGA, since it reflects the relation between goals and outcomes as main constituents of the control problems. Therefore it should have the most significant contribution to \hat{r} , while the cost terms $\hat{e}_c(\mathbf{c})$ and $\hat{e}_a(\mathbf{a})$ should rather be residuals. Hence, we can choose \mathbf{v} and \mathbf{s} such that $\hat{e}_c(\mathbf{c})$ and $\hat{e}_a(\mathbf{a})$ are minimal, which we can formulate with l_2 norms of the matrices \mathbf{R}_c and \mathbf{R}_a :

$$\mathbf{v}^*, \mathbf{s}^* = \underset{\mathbf{v}, \mathbf{s}}{\operatorname{argmin}}(E_R) = \underset{\mathbf{v}, \mathbf{s}}{\operatorname{argmin}}(\|\mathbf{R}_c\|_2 + \|\mathbf{R}_a\|_2)$$

This formulation involves the high-dimensional matrices \mathbf{R}_c and \mathbf{R}_a , but can be boiled down to an equivalent lower-dimensional problem. Inserting the definitions of \mathbf{H} , \mathbf{M} based on \mathbf{v} and \mathbf{s} gives

$$\begin{aligned}\mathbf{R}_c &= \mathbf{K}_{c,c} + \mathbf{U}'\mathbf{S}'\mathbf{v}\mathbf{s}^{-2}\mathbf{v}^T\mathbf{S}'\mathbf{U}'^T \quad \text{and} \\ \mathbf{R}_a &= \mathbf{K}_{a,a} + \mathbf{V}'\mathbf{v}\mathbf{s}^2\mathbf{v}^T\mathbf{V}'^T.\end{aligned}$$

\mathbf{v} and \mathbf{s} can only affect \mathbf{R}_c and \mathbf{R}_a within the subspace spanned by \mathbf{U}' and \mathbf{V}' . Hence, we can equivalently consider their n -dimensional projections

$$\begin{aligned}\mathbf{R}_c^{(n)} &= \mathbf{U}'^T\mathbf{K}_{c,c}\mathbf{U}' + \mathbf{S}'\mathbf{v}\mathbf{s}^{-2}\mathbf{v}^T\mathbf{S}' \quad \text{and} \\ \mathbf{R}_a^{(n)} &= \mathbf{V}'^T\mathbf{K}_{a,a}\mathbf{V}' + \mathbf{v}\mathbf{s}^2\mathbf{v}^T,\end{aligned}$$

and minimize the respective error, such that

$$\mathbf{v}^*, \mathbf{s}^* = \underset{\mathbf{v}, \mathbf{s}}{\operatorname{argmin}}(E_R^{(n)}) = \underset{\mathbf{v}, \mathbf{s}}{\operatorname{argmin}}(\|\mathbf{R}_c^{(n)}\|_2 + \|\mathbf{R}_a^{(n)}\|_2).$$

We are currently not aware of a closed form solution to this optimization problem. It is low-dimensional, though. In practice we found that it can be solved both efficiently and very effectively by simple gradient descent. Therefore we initialize $\mathbf{v} = \mathbf{s} = \mathbf{1}_n$ and apply gradient descent on $E_R^{(n)}$ with a step width of 0.001 until numeric convergence. Thereby only the diagonal values of \mathbf{s} are considered, and \mathbf{v} is orthonormalized by setting its singular values to 1 after each update.

C. Algorithm Summary and Interpretation

Altogether, LGA starts with a universal approximation of the reward or value (i.e. future expected reward) function in the quadratic form shown in equation 5. This mechanism corresponds to value systems supposed to exist in midbrain structures [32], [42], whereas the reward itself could reflect any extrinsic or intrinsic phenomenon.

The second step is the SVD of $\mathbf{K}_{c,a}$. Here we select the axis in column and row space that have the highest singular values. This corresponds to the axis of inside the action- and context space that are most significant to the reward/value function. Hence, this step identifies the low-dimensional observation space in which goals and action outcomes are situated.

The third and last step is to actually locate goals and action outcomes inside the observation space. Therefore we assemble the matrices \mathbf{H} and \mathbf{M} for goal- and self-detection out of the terms \mathbf{v} and \mathbf{s} . This directly gives the functions $\hat{h}(\mathbf{c})$ and $\hat{f}(\mathbf{a})$ and allows to compute $\hat{e}_c(\mathbf{c})$ and $\hat{e}_a(\mathbf{a})$ if needed.

We have argued on a purely conceptual level that goals should be considered as abstractions of reward signals. Latent Goal Analysis provides the computational tool for this

idea. Thereby goals and outcomes are considered latent variables of the reward function, which allows to view them as compact low-dimensional representations of otherwise high-dimensional actions and world states. These representations thereby do justice to the semantics of desire and intention by expressing aspects relevant to reward, and do justice to the achievement semantics by representing both in a common space in which they are compared. In the following experiments we will exemplify all of these aspects, and the universality of the approach, by showing (i) a dimension reduction application based on external rewards in Sec. IV, and (ii) a developmental study with generic and intrinsic information seeking rewards in Sec. V in which highly interpretable representations are learned, and used to bootstrap self-supervised motor learning.

IV. EXPERIMENT: DIMENSION REDUCTION IN NEWS ARTICLE RECOMMENDATION

Our computational learning formulation shows that goals can be learned as abstractions of rewards. The algorithm itself is in the first place meant to show the feasibility of this approach. However, since the formulation is based on spectral decomposition we can immediately apply it for *dimensionality reduction* of reward-based learning problems. Our first experiment therefore investigates this ability, for which we take a very *practical* scenario. Here, the “task” is in fact already given by means of extrinsic rewards. We will show that learned goals can serve as very useful and compact representations of such externally given tasks. We will contrast this external specification in Sec. V where we investigate purely intrinsic rewards that do not immediately describe any task.

Our experiment investigates LGA’s capability for dimension reduction in a one-step reinforcement learning problem. This scenario considers a website comprising a certain set $A(t)$ of news articles at each time. One article can be featured, i.e. *recommended*, at a prominent position on the website. The task is to select which article’s teaser (action $\mathbf{a} \in A(t)$) should be on the featured position. A *recommender system* is supposed to select these actions such that the probability that the website visitor interacts with it (e.g. clicks on the teaser, or purchases a good) is maximized, such that the earning of the operating web company rise in the end. In order to perform such selection specific to the visitor, there is in many cases information like country, age, or previous click-history (context \mathbf{c}) available due to IP-address, cookies, and a login at the website. With such information a reward function $r(\mathbf{c}, \mathbf{a})$ can be estimated that resembles the click probability. Dimensionality reduction, however, is crucial in this domain: both context and action are typically very high-dimensional, but any recommender system must react extremely quickly to thousands or millions of visits of a webpage. This can only be achieved if the dimension of \mathbf{c} and \mathbf{a} is reduced to allow for an efficient evaluation of $r(\mathbf{c}, \mathbf{a})$.

Dimension reduction in reinforcement learning is a tedious issue. One attempt has been to learn reduced rank regression of transition probabilities to guide exploration [43], but which can not directly consider the features’ *relevance* to reward achievement. Purely unsupervised schemes like PCA, or slow-feature analysis [44] are frequently applied for state-space

dimension reduction, but also cannot account for the actual reward-relevance. Reward-modulated versions of such learning rules [45] can account for the reward-relevance at least to some extent, but are limited to simple correlations. None of these methods can effectively reduce the dimension of *actions* because actions do not have a naturally observable probability distribution (except when expert demonstrations are given [46]). The only models that can consider states, actions, and reward at the same time estimate the reward function based on bi-linear regression and reduce the rank of the parameter matrix [47], [48], [49]. From a perspective of dimension reduction only, these models are similar to our approach, although coming from an entirely different direction. We will show experimentally that our method yields significantly more compact representations in a practical scenario [50].

A. Material and Method

For this experiment we use the “Yahoo! Front Page Today Module User Click Log Dataset, version 2.0” [50], which comprises recordings of the click behavior on *yahoo.com*’s front page from 15 consecutive days in October 2011, from which we utilize the first day of recordings only. This recording contains a total number of $T = 1.607.525$ events. Each event contains the actually displayed news-teaser, the set of currently available news, a set of visitor features, and the visitor’s decision to click on the teaser ($r = 1$) or not ($r = 0$). 49 different teasers have been shown in this period, which we represent as $m = 49$ dimensional actions \mathbf{a} encoded with a “1-of- m ” scheme. The events contain a total number of 116 binary features about the visitor, which are fully *anonymized* in the data, i.e. their individual meanings are not revealed by Yahoo. Using this data, we estimate \mathbf{K} using batch-gradient descent on the empirical error $E[(\hat{r}(\mathbf{c}, \mathbf{a}, \mathbf{K}) - r)^2]$. We applied 10000 epochs of training with a gradient step width 0.01 starting from zero initial parameters. After that the parameters were fine-tuned by applying a whitening on the contexts and continuing batch regression for another 1000 epochs with step width 0.0001.

As a baseline, we applied a bi-linear regression model $\hat{r}(\mathbf{c}, \mathbf{a}) = \psi_c(\mathbf{c})^T \cdot \mathbf{B} \cdot \psi_a(\mathbf{a})$ that was trained with the same procedure as the quadratic model. Such bi-linear regression (BLR) models have previously [47], [48] been used to reduce the dimension in recommender scenarios: the matrix \mathbf{B} can be decomposed into $\mathbf{U}_B \mathbf{S}_B \mathbf{V}_B^T$ by singular value decomposition, after which only the n most significant dimensions are kept. This decomposition allows to rewrite \hat{r} as

$$\hat{r}(\mathbf{c}, \mathbf{a}) = \left(\mathbf{S}_B^{\frac{1}{2}} \mathbf{U}_B^T \cdot \psi_c(\mathbf{c}) \right)^T \cdot \left(\mathbf{S}_B^{\frac{1}{2}} \mathbf{V}_B^T \cdot \psi_a(\mathbf{a}) \right)$$

Very similar to the LGA approach, this method can be interpreted as projecting contexts and actions into a common space, which has been referred to “partworth” space [49]. In this space, the comparison is done by a scalar product, whereas LGA measures a direct distance. The “partworth” space therefore does not encode *goals* because they cannot be definitely achieved (the scalar product is not bounded). The evaluation cost for both models is the same: both involve

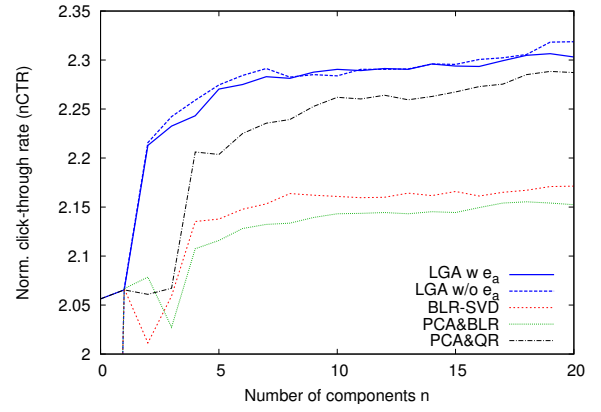


Fig. 3: Results for the Yahoo! Webscope R6B data set. Estimated normalized click-through rates for LGA, BLR decomposition, and PCA depending on the number of selected components.

matrix-vector multiplications of the same size; the square-distance in LGA is only exactly one floating point operation more expensive than the scalar product. As further baselines we used PCA to reduce the dimension of the context-space before applying either quadratic or bi-linear regression. In the PCA condition the dimension of actions cannot be reduced.

B. Results

In order to evaluate the effectiveness of each technique we need a domain specific evaluation metric, i.e. how many visitors’ clicks both approaches can generate. For each method we can denote the policy to choose a news-teaser \mathbf{a} based on the user information \mathbf{c} as $\pi(\mathbf{c}) = \operatorname{argmax}_{\mathbf{a} \in A(t)} \hat{r}(\mathbf{c}, \mathbf{a})$, where $A(t)$ is the set of articles available at the time t of the page visit. A natural performance metric then is the *click-through rate* $\text{CTR}_\pi = N_\pi^+ / N$, where N is the total number of page visits and N_π^+ is the number of clicks generated by selecting teasers with π . Yet, this measure can only be thoroughly measured when the policy is run *online* on the webpage. For an *offline evaluation* [49] we can estimate the performance by counting how often an actually clicked teaser would have been also recommended by the policy:

$$\text{nCTR} = \frac{\text{CTR}_\pi}{\text{CTR}_\%} \approx \frac{|\{r_t = 1 \wedge \mathbf{a}_t = \pi(\mathbf{c}_t)\}|}{\sum_t (r_t \cdot |A(t)|^{-1})},$$

which is baselined against the performance $\text{CTR}_\%$ of a uniform random strategy. Fig. 3 shows that LGA achieves a substantially better performance than the bi-linear model decomposition (BLR-SVD). In fact, already $n = 0$ components for LGA, which corresponds to only evaluating $\hat{e}_a(\mathbf{a})$, suffice to reach a factor of 2.06 compared to chance level. This reflects the simple fact that some articles are more interesting than others. LGA then quickly improves to $\text{nCTR} > 2.25$ for $n \geq 5$ components and further improves to $\text{nCTR} > 2.3$. The bi-linear model requires $n \geq 8$ components to reach only $\text{nCTR} > 2.15$ with only minimal further improvement for more components. It might be considered unfair to allow the $\hat{e}_a(\mathbf{a})$ term for LGA, whereas the bi-linear models do neither comprise nor require such an additional measure. In the recommender scenario, however, it is reasonable because

$\hat{e}_a(\mathbf{a})$ only needs to be computed *once* when an article is published. The cost for evaluating the policy on a page visit is the same, since both LGA and BLR need to multiply the visitor context with a $(n \times p)$ matrix. LGA then computes a distance in n dimension and BLR a scalar product in n dimensions, which also has the same cost. However, in this particular setup LGA achieves equally high performance (for $n > 0$) when the cost term is omitted (see Fig. 3). Both LGA and bi-linear decomposition outperform their counterparts with unsupervised PCA on the states before running bi-linear (BLR) or quadratic (QR) regression. Interestingly, PCA&QR substantially outperforms the standard BLR decomposition approach, which shows the high expressiveness of quadratic regression in general. PCA&QR has, however, an increased cost of evaluating $\hat{r}(\mathbf{c}, \mathbf{a})$ for a new context: $O(n^2 + n \cdot p)$ instead of $O(n + n \cdot p)$ for all other methods. LGA still extracts more compact representations from the quadratic regressor and outperforms PCA&QR.

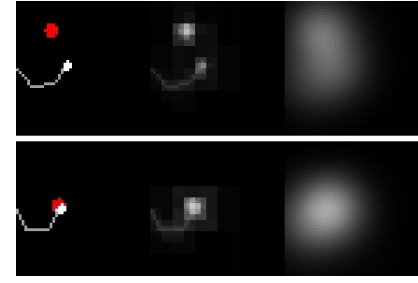
C. Discussion

We can conclude that LGA allows for an effective dimensionality reduction in the online news recommender setup, in which it outperforms the standard bi-linear model in terms of generated clicks. The margin is thereby numerically not very high in the range of 5-10%, but which is still highly significant to the domain since clicks are directly related to a website’s monetary income. Other studies on recommender systems have reported much higher absolute values of CTR for other benchmarks, which suggests that the data set used here is rather hard. A possible reason is that there are no features for the actions, but only identities. Features to relate similar articles could therefore further increase both the absolute CTR as well as the margin between different methods.

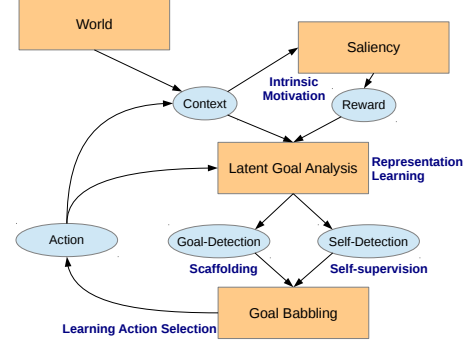
Goals seem to be useful abstractions in this setup. Within our framework goals and action outcomes serve as low-dimensional, compact representations of what defines the reward. In this particular case, though, it is impossible to interpret the goals’ semantics, simply for the reason that we do not know the meaning of the context features and the content of the articles, all of which had been previously anonymized. If we *would* know them, however, we could interpret the learned observation space as a space of different *topics of interest* of users. From the webserver’s perspective the user (its context \mathbf{c}) comes with a topic of interest that is described as *goal* in this space. The webserver has to-be-shown articles as actions which are described in the topic space by means of action outcomes: the “result” of choosing a particular article is to hit a certain point in the interest space. The achievement semantics describe that if both match, the user is *interested* and likely to generate a reward (click) to the system.

V. EXPERIMENT: GOAL DEVELOPMENT IN REACHING

In the last section we have shown that LGA can extract goals as useful abstractions from extrinsic rewards which already specify a concrete task. But how could an agent learn goals when the task is not that explicitly or not at all given already in the beginning? In this section we contrast the previous



(a) Arm image, saliency, smoothed saliency



(b) Schematic organization of the experiment

Fig. 4: We consider saliency to generate an information seeking reward, from which LGA learns representations that are used by Goal Babbling.

experiment with *intrinsic rewards* that do not describe any particular task. We consider visual saliency as a reward for a simple robot arm to implement *information seeking* behavior [31] and show that it leads to meaningful goal- and self-representations. Saliency measures have already been shown to permit a self-detection of the own end-effector [51], simply because looking at the own hand is “interesting”. Here we extend this finding by considering an object at the same time. It turns out that more interesting than looking at the hand *or* the object is to look at *both* closely together (compare Fig. 5a(b) top and bottom). We show that LGA thereby develops a detection of an external object as goal, and a self-detection of the own hand. These representations are thereby already *utilized* by means of goal babbling [4] in a closed loop, which results in the *emergence of goal-directed reaching*.

A. Setup

The basic scenario is shown in Fig. 4. We consider a simple robot arm with three joints (segment length $1/3$ each), such that *actions* are the joint angles $\mathbf{a}_t \in \mathbb{R}^3$. We refer to the effector’s actual position (that is at no time explicitly known the learner) in cartesian coordinates as $x_t \in \mathbb{R}^2$. A salient object is placed somewhere in the scene at coordinates $o_t \in \mathbb{R}^2$. Arm and object together are rendered into a 48×48 pixel image. Generically we could think of this image as *context* in terms of visual perception. However, considering raw 2300 dimensional visual input for learning is neither computationally feasible nor very biologically plausible. For this experiment we assume a certain extent of image processing that has already identified the object and hand coordinates as keypoints in this image. The context for learning comprises these basic coordinates plus

additional noise dimensions to challenge learning. At every timestep t the agent is assumed to be still in position x_{t-1} , with the object at position o_t . Hence, the learner's context is $\mathbf{c}_t = (o_t; x_{t-1}; \varepsilon) \in \mathbb{R}^6$ with gaussian noise $\varepsilon \in \mathbb{R}^2, \varepsilon_i \sim \mathcal{N}(0.5, 0)$. For the to be estimated functions \hat{h} , \hat{f} , \hat{e}_c , and \hat{e}_a we use a locally-linear learning formulation identical to [19] with receptive field radii 2.0, 0.25, 0.5, and 0.5 respectively. As a design choice we selected $n=2$ components to be extracted from the 6 dimensional context and 3 dimensional action.

Reward: The learner's reward r_t is computed using a simple saliency model based on difference-of-gaussians (simplified from Itti's model [52]). After the agent has received the context \mathbf{c}_t (containing an image of the old action \mathbf{a}_{t-1}) and selected a new action \mathbf{a}_t , we compute a reward based on the "after-action" image containing the object and the new action \mathbf{a}_t . Then, we compute a pyramid of gaussians: The image is smoothed with a 5x5 gaussian kernel and scaled down by a factor of 2. This procedure is repeated 4 times. The saliency map (Fig. 4(a), middle) is computed out of these 5 images (original & 4x smoothed) by taking the the difference between any two of them, and adding up the amplitudes of those differences. Considering the most salient point would mean to look at the most interesting pixel. However, we assume that the agent does not just attend to a single visual receptor but rather a *region* in the visual scene. Therefore we smooth the saliency map on a large scale (Fig. 4(a), right) with a gaussian filter with $\sigma = 10$ pixels width which models the total width of the agent's visual field. The *highest value* of this *smoothed saliency* therefore measures how much information the agent can have in its visual field. We manually normalized the scale of these values such that they approximately lie in a range $[0; 1]$ and consider these the rewards r_t .

Online Learning Algorithm: For this experiment we developed an *online* algorithm for LGA. While the previously introduced algorithm is useful to show the theoretic feasibility of LGA and for batch processing, it is not suited for the closed loop processing intended in this experiment. Here we use a simple gradient descent algorithm to estimate the goal- and self-detection *directly* (as opposed to learning a full reward $r(\mathbf{c}, \mathbf{a})$ model first). We consider an agent that observes samples along time t . The agent perceives a context \mathbf{c}_t , executes an action \mathbf{a}_t , and receives a reward $r_t = r(\mathbf{c}_t, \mathbf{a}_t)$ based on a hidden reward function $r(\mathbf{c}, \mathbf{a})$. The agent is supposed to learn the functions \hat{h} , \hat{f} , \hat{e}_c , and \hat{e}_a such that the observed reward r_t is explained by them according to Eqn. 3. This can be done by reducing the *reward-prediction error*:

$$E_t(r_t, \mathbf{c}_t, \mathbf{a}_t) = ||e_t(r_t, \mathbf{c}_t, \mathbf{a}_t)||^2 = ||r_t - \hat{r}(\mathbf{c}_t, \mathbf{a}_t)||^2.$$

We denote the learnable parameters of \hat{h} , \hat{f} , \hat{e}_c , and \hat{e}_a as θ_h , θ_f , θ_c and θ_a respectively. For an initial symmetry-breaking (due to the invariance of internal rotation and translation) it is necessary to initialize θ_h and θ_f with small random values. From this point on, simple gradient descent on E can succeed to estimate the functions. However, we need to consider that the values of \hat{e}_c and \hat{e}_a have to be kept small. For this purpose we use a simple decay term similar to weight-decay often used in neural networks: parameter values are not only adapted by the error-reduction signal, but also by a decay of some $\epsilon \in [0; 1]$

portion of their value. Since any reward mass that decays from \hat{e}_c and \hat{e}_a needs to be explained by $||\hat{h} - \hat{f}||$ instead, we *add* (reverse sign) the decay values to the learning signals of \hat{h} and \hat{f} . The resulting gradient rule with rates η_d and η_c is:

$$\begin{aligned} \Delta\theta_f &= +\eta_d \cdot (e_t + \epsilon \cdot [\hat{e}_a(\mathbf{a}_t) + \hat{e}_c(\mathbf{c}_t)]) \cdot (\hat{\mathbf{x}}_t^* - \hat{\mathbf{x}}_t) \cdot \frac{\partial \hat{f}(\mathbf{a}_t)}{\partial \theta_f} \\ \Delta\theta_h &= -\eta_d \cdot (e_t + \epsilon \cdot [\hat{e}_a(\mathbf{a}_t) + \hat{e}_c(\mathbf{c}_t)]) \cdot (\hat{\mathbf{x}}_t^* - \hat{\mathbf{x}}_t) \cdot \frac{\partial \hat{h}(\mathbf{c}_t)}{\partial \theta_h} \\ \Delta\theta_a &= \eta_c \cdot (e_t - \epsilon \cdot \hat{e}_a(\mathbf{a}_t)) \cdot \frac{\partial \hat{e}_a(\mathbf{a}_t)}{\partial \theta_a} \\ \Delta\theta_c &= \eta_c \cdot (e_t - \epsilon \cdot \hat{e}_c(\mathbf{c}_t)) \cdot \frac{\partial \hat{e}_c(\mathbf{c}_t)}{\partial \theta_c}, \end{aligned}$$

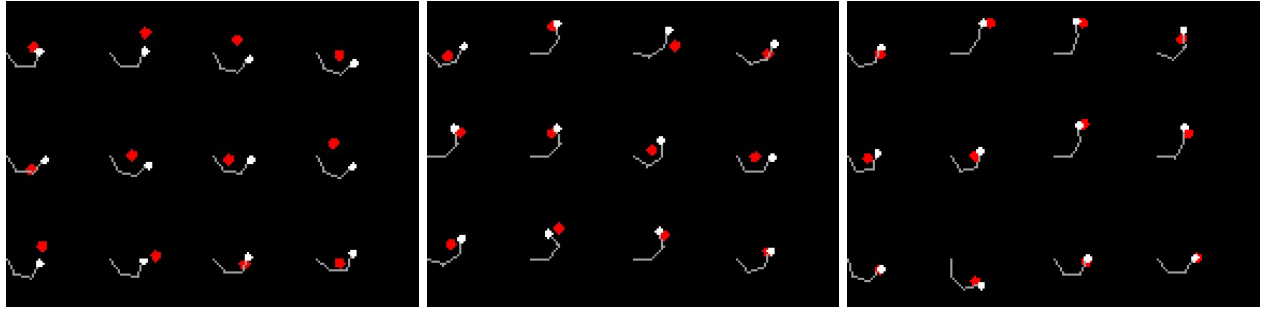
in which the last term in each formula is depending on the function approximation method. If the decay term is disabled ($\epsilon=0$), these formulas correspond to ordinary gradient descent on E . The decay term balances the contribution of all terms such that goal- and self-detection take the dominating role in explaining the reward. The term $-||\hat{h} - \hat{f}||^2$ can, however, not model arbitrary reward functions alone. In particular, this negative distance can only account for *numerically* negative rewards. Modeling numerically positive rewards requires the terms \hat{e}_c and \hat{e}_a to shift the entire estimate \hat{r} by a constant. If the decay term is used this process can never fully reach the necessary shift. In order to still permit a reasonable learning signal for \hat{h} and \hat{f} , we introduce a new and purely scalar term k into the reward estimation. This term is not effected by the decay, but can shift the reward estimate such that $-||\hat{h} - \hat{f}||^2$ can be used to model the shape of the reward function:

$$\begin{aligned} \hat{r}(\mathbf{c}, \mathbf{a}) &= -||\hat{h}(\mathbf{c}) - \hat{f}(\mathbf{a})||^2 + \hat{e}_c(\mathbf{c}) + \hat{e}_a(\mathbf{a}) + k \\ \Delta k &= \eta_c \cdot e_t \end{aligned}$$

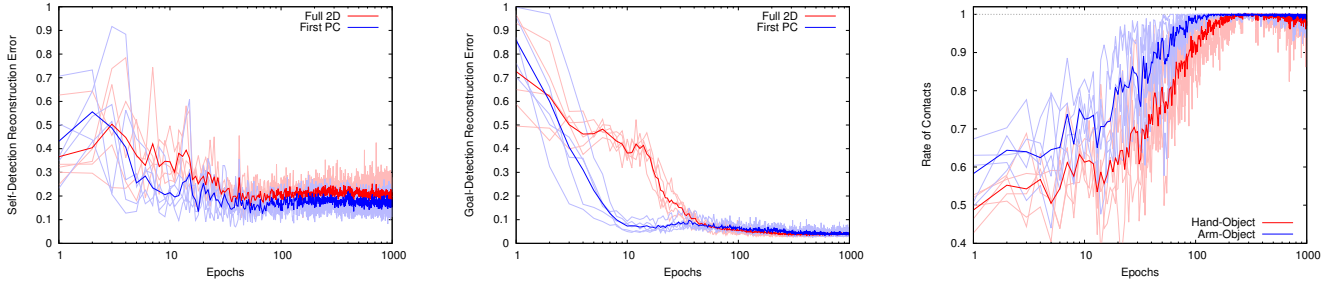
In this experiment we utilize only \hat{h} and \hat{f} , whereas \hat{e}_a could potentially be used to select cost-optimal actions for the same goal. The term \hat{e}_c (and k) is not directly useful, but needs to accompany the estimation when approximating any possible reward function as shown in Sec. III.

Procedure: We conducted this experiment with 5 independent trials with $t=10^6$ samples each. During a continuous movement of the object in the visual scene we performed continuous online learning of the LGA with learning rates $\eta_d = 0.015$, $\eta_c = 0.005$, and $\epsilon = 0.05$. While the entire procedure is possible in an online fashion only, we decided to perform an additional consolidation phase to speed up learning. Therefore the generation of new samples is interrupted every epoch of 1.000 samples, and the last 10.000 samples are presented in random ordering 10 more times.

LGA describes how an agent can learn internal representations of goals and self. It does *not* instantaneously describe a strategy to select actions. However, we can use the goal- and self-detection to perform self-supervised motor learning: the executed actions \mathbf{a}_t and estimated outcomes $\hat{\mathbf{x}}_t = \hat{f}(\mathbf{a}_t)$ allow to generate a supervised learning signal for common methods of motor learning [3], that can be used together with the estimated goal $\hat{\mathbf{x}}_t^* = \hat{h}(\mathbf{c}_t)$ to perform goal-directed motor-control. Here we utilize a previous algorithm for *goal babbling* [19], that also utilizes the goals in order to scaffold learning.



(a) During learning, the system develops from initially no coordination (left) to 1d reaching (middle) to 2d reaching (right).



(b) The self-detection encodes the hand position. (c) The goal-detection encodes the object position. (d) Hand-object contact rate via the inverse model.

Fig. 5: Goal- and self-detection learn to represent the object and hand position respectively. The first principal component in the internal coordinate system encodes the top-down coordinate. Goal babbling uses these abstractions to learn an inverse model that consistently makes contact with the object.

This algorithm learns an *inverse model* $g: \hat{\mathbf{x}}^* \mapsto \mathbf{a}$ from self-generated examples $(\hat{\mathbf{x}}_t, \mathbf{a}_t)$. The actions are selected by trying to accomplish the goal by means of the inverse model plus some exploratory noise N_t : $\mathbf{a}_t = g(\hat{\mathbf{x}}_t^*) + N_t(\hat{\mathbf{x}}_t^*)$. For this we use a learning rate 0.02, local model distances 0.15 and exploratory noise with amplitude 0.15 (see [19]).

The entire organization of the experiment is shown in Fig. 4(b). The world provides an object that gets encoded in the context together with the agent’s last action. The agent’s saliency system generates an intrinsic reward for information seeking. Latent Goal Analysis extracts the reward-relevant information from the action (self-detection) and disentangles the goal from other information in the context in order to explain the reward by the relation between goal and self. Self-detection and estimated goals are then used for self-supervised goal babbling in a closed loop.

B. Results

We ran an evaluation of every 1.000 samples between two consolidation steps. We investigated three questions: (i) What does the self-detection encode? (ii) What does the goal-detection encode? (iii) What behavior results from that abstractions? In order to investigate (i) and (ii) we checked how well the *internal* representations of outcomes $\hat{\mathbf{x}}$ and goals $\hat{\mathbf{x}}^*$ describe values of the *actual* effector position \mathbf{x} and object position \mathbf{o} . Even if the internal variables encode them perfectly there can be arbitrary shifts and translations in the internal coordinate system. Therefore we computed the best linear fit L from internal representations to actual variables. We assessed the quality of the encoding by the normalized root-mean-square error (NRMSE) $\sqrt{E[||L_x(\hat{\mathbf{x}}) - \mathbf{x}||^2]} / \sqrt{\text{Var}[\mathbf{x}]}$ (correspondingly for $\hat{\mathbf{x}}^*$ and \mathbf{o}). If this error is 0, the value of

the actual variable can be perfectly (linearly) predicted from the internal one: the internal representation encodes the actual variable. A value of 1.0 means that the prediction gives an error in the range of the variable’s variance, which indicates that the internal variable does not encode the actual one at all.

Results for the self-detection are shown in Fig. 5b. Learning a representation of the robot’s own hand requires a strongly non-linear multi-dimensional mapping. Results show that already in the very beginning there seems to be a certain extent of encoding with errors significantly below 1.0. However, this results merely from the low versatility of actions in the beginning. Goal babbling initially chooses actions close to a single posture since it is not sufficiently trained yet. The outcomes of such locally distributed postures can to a limited extent be predicted with the randomly initialized self-detection. After approx. 50 epochs the values stabilize around 0.2 which means that 80% of the actual effector-position’s variance can be explained by the internal representations. At later stages there is a minimal increase of the error values which is because goal babbling learns to use more and more different and widespread postures. Hence, the population gets less local and harder to describe due to non-linearities. Latent Goal Analysis after all succeeds to learn the robot arm’s forward function from joint angles to effector position by just using the saliency reward. We additionally investigated the encoding by checking what different coordinate axes in the learned representation encode. The blue line in Fig. 5b shows the prediction of the effector’s top/down coordinate from just the highest variance principle component of the internal representation. Low errors indicate that this axis indeed encodes top/down movements.

If LGA should learn a goal representation that describes goal-directed reaching, then the extracted goals $\hat{\mathbf{x}}^*$ should encode the object position \mathbf{o} . This could seem simple because

the object location is already directly encoded in the context c . However, this variable still needs to (i) be identified as the relevant one among other entries (noise and previous hand-position) in the context, and (ii) set into the right relation (i.e. orientation, shift, scaling) to the self-detection. In particular at later stages of learning in our experiment the own hand-position strongly correlates with the object position due to goal-directed reaching, so that keeping track of the right variable is far from trivial. Results in Fig. 5c show that at the time of initialization the object position is not at all encoded in the goal-detection. Then, the strongest principal component of \hat{x}^* quickly coincides with the top-down axis of the object position (blue lines). Few epochs later, the goals' 2D values (red lines) do largely encode the actual object position with errors around 0.05.

Results so far show that the robot's hand position and the object position are indeed found as representations for self and goals. In order to check how well they fit together (i.e. whether they are in the right geometric relation inside the observation space), we checked the behavior generated by goal babbling as a result of both abstractions. In order to perform an analysis that excludes exploratory noise (to check the representations themselves) we evaluate the combination of goal-detection \hat{h} and inverse model g (learned based on \hat{f}). The function $g(\hat{h}(c))$ suggests actions a for any context c . Hence we can check those actions and see whether they correspond to a reaching act towards the object position encoded in c . We counted for the contexts within each epoch how often the actions led to a contact of either hand and object, or the whole arm and the object (based on the geometries and sizes in Fig. 4(a)). Results in Fig. 5d show that learning rapidly seeks for contacts of arm and object first, and shortly later establishes a 100% contact rate of the robot's hand and the object. Example performances shown in Fig. 5a thereby give an explanation for the not perfectly accurate relation between self-detection and actual effector position despite 100% contact rate: There is not need to encode the "tool-center" position of the hand – as it is usually done in robotics due to the lack of more adaptive representations. Rather, our model learns to contact objects close to the shoulder with the inner part of the effector, while far away objects are contacted with the outer part of the effector.

C. Discussion

Latent Goal Analysis together with goal babbling indeed produces representations as well as inverse models that correspond to goal directed reaching in this experiment. Remarkably, all of this is bootstrapped from a task-unspecific reward based on visual saliency as a sole original learning signal. Abstractions bootstrapped by LGA are then used as *self-supervised* learning signal for goal babbling. This experiment therefore has a dual finding: Firstly, we show that saliency as an information seeking reward results in goal-directed reaching behavior as seen from a *distal perspective*. While it had been reported that saliency can account for self-detection [51] of the own hand, we are not aware of studies also showing goal-directed behavior as a direct consequence. Secondly, we

find that this information seeking reward *together* with Latent Goal Analysis can account for the *emergence of internal self and goal representations* (proximal perspective) as they are typically preprogrammed in robotics, and pre-supposed in computational neuroscience of internal models.

VI. DISCUSSION

The autonomous development of goals is a fundamental issue in developmental robotics. This paper has proposed a detailed conceptual framework and a mathematical operationalization for agents to learn goals themselves. Our main argument therefore is to consider goals as high level *abstractions* of lower level mechanisms of intention such as reward and value systems which themselves could origin from either external rewards or intrinsic motivation measures. We emphasized the need to consider and learn goals alongside with self-detection of the own actions' outcomes. Both can then be compared in a common space. We suggest that goals and outcomes together can be learned by considering them as latent variables (i.e. abstractions) that can explain an observed or expected future intrinsic or external reward. Our computational approach Latent Goal Analysis operationalizes this thought. We have therefore shown the universal existence of the transformation from rewards to goals.

We have shown two very different experiments with this learning formulation: On the one hand a concrete task that is already specified by external rewards, and for which LGA can learn goals as very compact and practically useful representations. On the other hand we have shown that considering mere visual saliency as a generic, task-unspecific information seeking reward to be processed by our framework leads to abstractions of self and goals, that ultimately lead to goal-directed reaching behavior. In doing so we have not only shown what those abstractions encode, but have already capitalized on them for self-supervised motor learning and goal babbling. We think that these very complementary two cases highlight the generality of our conceptual and mathematical approach. Of course, it will need more examples in future research to substantiate this claim. For that purpose, we think that our work does indeed widely open the door for further investigations for instance about social learning scenarios [13] or also other measures of intrinsic motivation [5], [53].

In particular the case of intrinsic rewards gives rise to interesting questions on a more general cognitive science side. In the case of our saliency experiment the robot does not start with any particular task semantics. Rather, its generic information seeking leads to goal-directed behavior. The novelty in our study is thereby that this behavior is generated and accompanied also by goal *representations* of what the agent is going for, which is often seen as the crucial feature of *intentional* action [54]. Are these goals the robot's "own" goals? We hope that our work stimulates discussion around this issue, and can provide more detailed answers of how infants become intentional agents having their *own* goals, and robots being able to have theirs, too.

REFERENCES

- [1] J. Bruce and M. Veloso, "Real-time randomized path planning for robot navigation," in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, vol. 3. IEEE, 2002, pp. 2383–2388.
- [2] W. Chung, L.-C. Fu, and S.-H. Hsu, "Chapter 6: Motion control," in *Handbook of Robotics*, B. Siciliano and O. Khatib, Eds. Springer New York, 2007, pp. 133–160.
- [3] D. Nguyen-Tuong and J. Peters, "Model learning for robot control: a survey," *Cognitive Processing*, vol. 12, no. 4, 2011.
- [4] M. Rolf, J. J. Steil, and M. Gienger, "Goal babbling permits direct learning of inverse kinematics," *IEEE Trans. Autonomous Mental Development*, vol. 2, no. 3, 2010.
- [5] A. Baranes and P.-Y. Oudeyer, "Active learning of inverse models with intrinsically motivated goal exploration in robots," *Robotics and Autonomous Systems*, vol. 61, no. 1, pp. 49–73, 2013.
- [6] M. Rolf, "Goal babbling with unknown ranges: A direction-sampling approach," in *IEEE Int. Joint Conf. Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, 2013.
- [7] D. M. Wolpert, R. C. Miall, and M. Kawato, "Internal models in the cerebellum," *Trends Cog. Science*, vol. 2, no. 9, pp. 338–347, 1998.
- [8] M. Ito, "Control of mental activities by internal models in the cerebellum," *Nature Reviews Neuroscience*, vol. 9, pp. 304–313, April 2008.
- [9] D. M. Wolpert, K. Doya, and M. Kawato, "A unifying computational framework for motor control and social interaction," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 358, no. 1431, pp. 593–602, 2003.
- [10] E. A. Locke and G. P. Latham, "Goal setting theory," *Motivation: Theory and research*, pp. 13–29, 1994.
- [11] H. Bekkering, A. Wohlschlaeger, and M. Gattis, "Imitation of gestures in children is goal-directed," *The Quarterly Journal of Experimental Psychology: Section A*, vol. 53, no. 1, pp. 153–164, 2000.
- [12] G. Gergely and G. Csibra, "Teleological reasoning in infancy: the naive theory of rational action," *Trends Cog. Science*, vol. 7, no. 7, pp. 287–292, 2003.
- [13] B. Wrede, K. Rohlfing, J. Steil, S. Wrede, P.-Y. Oudeyer, and J. Tani, "Towards robots with teleological action and language understanding," in *Humanoids 2012 Workshop on Developmental Robotics: Can developmental robotics yield human-like cognitive abilities?*, 2012.
- [14] J. Weng, J. McClelland, A. Pentland, O. Sporns, I. Stockman, M. Sur, and E. Thelen, "Autonomous mental development by robots and animals," *Science*, vol. 291, no. 5504, pp. 599–600, 2001.
- [15] C. Prince, N. Helder, and G. Hollich, "Ongoing emergence: A core concept in epigenetic robotics," in *Int. Conf. Epigenetic Robotics*, 2005.
- [16] S. Schaal, "Is imitation learning the route to humanoid robots?" *Trends in cognitive sciences*, vol. 3, no. 6, pp. 233–242, 1999.
- [17] M. Muhlig, M. Gienger, J. J. Steil, and C. Goerick, "Automatic selection of task spaces for imitation learning," in *IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 2009, pp. 4996–5002.
- [18] A. Y. Ng, S. J. Russell et al., "Algorithms for inverse reinforcement learning," in *ICML*, 2000, pp. 663–670.
- [19] M. Rolf, J. J. Steil, and M. Gienger, "Online goal babbling for rapid bootstrapping of inverse models in high dimensions," in *IEEE ICDL-EpiRob*, 2011.
- [20] G. T. Doran, "There's a S.M.A.R.T. way to write management's goals and objectives," *Management Review*, vol. 70, no. 11, pp. 35–36, 1981.
- [21] J. J. Gibson, "The theory of affordances," in *Perceiving, Acting, and Knowing*, R. Shaw and J. Bransford, Eds., 1977, pp. 67–82.
- [22] E. Sahin, M. Cakmak, M. Dogar, E. Ugur, and G. Ucoluk, "To afford or not to afford: A new formalization of affordances toward affordance-based robot control," *Adaptive Behavior*, vol. 15, no. 4, 2007.
- [23] J. K. O'Regan et al., "What it is like to see: A sensorimotor theory of perceptual experience," *Synthese*, vol. 129, no. 1, pp. 79–103, 2001.
- [24] Y. Nagai, A. Nakatani, S. Qin, H. Fukuyama, M. Myowa-Yamakoshi, and M. Asada, "Co-development of information transfer within and between infant and caregiver," in *IEEE Int. Conf. Development and Learning and Epigenetic Robotics (ICDL)*, 2012, pp. 1–6.
- [25] B. Hommel, "Action control according to TEC (theory of event coding)," *Psychological Research PRPF*, vol. 73, no. 4, pp. 512–526, 2009.
- [26] S. Verschoor, M. Weidema, S. Biro, and B. Hommel, "Where do action goals come from? evidence for spontaneous action-effect binding in infants," *Frontiers in Psychology*, vol. 1, no. 201, pp. 1–6, 2009.
- [27] R. M. Ryan and E. L. Deci, "Intrinsic and extrinsic motivations: Classic definitions and new directions," *Contemporary educational psychology*, vol. 25, no. 1, pp. 54–67, 2000.
- [28] J. Schmidhuber, "Formal theory of creativity, fun, and intrinsic motivation (1990-2010)," *IEEE Trans. Autonomous Mental Development*, vol. 2, no. 3, pp. 230–247, 2010.
- [29] P.-Y. Oudeyer, F. Kaplan, and V. V. Hafner, "Intrinsic motivation systems for autonomous mental development," *IEEE Trans. Evolutionary Computation*, vol. 11, no. 2, pp. 265–286, 2007.
- [30] G. Baldassarre, "What are intrinsic motivations? a biological perspective," in *IEEE Int. Joint Conf. Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, 2011.
- [31] J. Gottlieb, "Attention, learning, and the value of information," *Neuron*, vol. 76, no. 2, pp. 281–295, 2012.
- [32] W. Schultz, P. Dayan, and P. R. Montague, "A neural substrate of prediction and reward," *Science*, vol. 275, no. 5306, 1997.
- [33] A. Edsinger and C. C. Kemp, "What can i control? a framework for robot self-discovery," in *Int. Conf. Epigenetic Robotics*, 2006.
- [34] A. Stoytchev, "Self-detection in robots: a method based on detecting temporal contingencies," *Robotica*, vol. 29, pp. 1–21, 2011.
- [35] M. Hoffmann, H. G. Marques, A. Hernandez Arieta, H. Sumioka, M. Lungarella, and R. Pfeifer, "Body schema in robotics: a review," *IEEE Trans. Autonomous Mental Development*, vol. 2, no. 4, pp. 304–324, 2010.
- [36] M. Rolf and J. J. Steil, "Explorative learning of inverse models: a theoretical perspective," *Neurocomputing*, vol. 131, pp. 2–14, 2014.
- [37] A. L. Strehl, "Associative reinforcement learning," in *Encyclopedia of Machine Learning*. Springer, 2010, pp. 49–51.
- [38] J. Langford and T. Zhang, "The epoch-greedy algorithm for contextual multi-armed bandits," in *NIPS*, 2008.
- [39] A. G. Barto, "Reinforcement learning in motor control," in *Handbook of Brain Theory and Neural Networks*. Cambridge: MIT Press, 1994, pp. 809–813.
- [40] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *arXiv preprint cs/9605103*, 1996.
- [41] A. G. Barto and T. G. Dietterich, "Reinforcement learning and its relationship to supervised learning," *Handbook of learning and approximate dynamic programming*. John Wiley and Sons, Inc, 2004.
- [42] N. D. Daw and K. Doya, "The computational neurobiology of learning and reward," *Current opinion in neurobiology*, vol. 16, no. 2, pp. 199–204, 2006.
- [43] A. Nouri and M. L. Littman, "Dimension reduction and its application to model-based exploration in continuous spaces," *Machine Learning*, vol. 81, no. 1, pp. 85–98, 2010.
- [44] R. Legenstein, N. Wilbert, and L. Wiskott, "Reinforcement learning on slow features of high-dimensional input streams," *PLoS Computational Biology*, vol. 6, no. 8, 2010.
- [45] I. Bar-Gad, G. Morris, and H. Bergman, "Information processing, dimensionality reduction and reinforcement learning in the basal ganglia," *Progress in neurobiology*, vol. 71, no. 6, pp. 439–473, 2003.
- [46] S. Bitzer, "Nonlinear dimensionality reduction for motion synthesis and control," 2011.
- [47] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *IEEE Computer*, vol. 42, no. 8, pp. 30–37, 2009.
- [48] W. Chu and S.-T. Park, "Personalized recommendation on dynamic content using predictive bilinear models," in *Int. Conf. World wide web (WWW)*, 2009.
- [49] W. Chu, S.-T. Park, T. Beaupre, N. Motgi, A. Phadke, S. Chakraborty, and J. Zachariah, "A case study of behavior-driven conjoint analysis on yahoo!: front page today module," in *Int. Conf. Knowledge Discovery and Data Mining*, 2009.
- [50] Yahoo! Webscope, "R6B - Yahoo! Front Page Today Module User Click Log Dataset, version 2.0." [Online]. Available: <http://webscope.sandbox.yahoo.com>
- [51] M. Hikita, S. Fuke, M. Ogino, T. Minato, and M. Asada, "Visual attention by saliency leads cross-modal body representation," in *IEEE Int. Conf. Development and Learning (ICDL)*, 2008.
- [52] L. Itti, C. Koch, and E. Niebur, "A Model of Saliency-Based Visual Attention for Rapid Scene Analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.
- [53] K. Merrick, "A comparative study of value systems for self-motivated exploration and learning by robots," *IEEE Trans. Autonomous Mental Development*, vol. 2, no. 2, pp. 119–131, 2010.
- [54] D. McFarland, "Opportunity versus goals in robots, animals and people," in *Comparative approaches to cognitive science*. MIT Press Cambridge, MA, 1995, pp. 415–433.